NPS55HK72021A

United States
Naval Postgraduate School

N JOB, ONE MACHINE SCHEDULING TO MINIMIZE THE NUMBER OF LATE JOBS WHEN SET-UP TIMES ARE SEQUENCE DEPENDENT

by

S. Balut

and

G. Howard

February 1972

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral A. S. Goodfellow, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT:

Two algorithms have been formulated for scheduling n jobs through a single facility to minimize the number of late jobs when set-up times are sequence dependent. The first is a simple matrix algorithm which solves the problem when jobs must be processes in first-come, first-served (FCFS) order. The second is a branch and bound technique which arrives at an optimal solution with no restrictions on the sequence used. Both algorithms are demonstrated by examples.

Prepared by:

N JOB, ONE MACHINE SCHEDULING TO MINIMIZE THE NUMBER OF LATE JOBS WHEN SET-UP TIMES ARE SEQUENCE DEPENDENT

I. INTRODUCTION 1

II. PROBLEM DESCRIPTION 1

III. JOBS PROCESSED IN FCFS ORDER 2

A. The FCFS Algorithm 3

B. Example 4

C. Proof of Optimality 7

D. Comparison to Moore's Algorithm 8

IV. NO RESTRICTION ON PROCESSING SEQUENCE 8

A. The Branch and Bound Algorithm 9

B. Example 10

C. Discussion 13

I. INTRODUCTION

The problem treated is the n job, one machine scheduling problem in which set-up times are sequence dependent. Algorithms are presented which minimize the number of late jobs for two variations of the problem.

In the first, jobs are restricted to be processed in first-come, first-served (FCFS) order. After presentation of this algorithm it is applied to an example problem and then proved to produce an optimal schedule. The following section draws a comparison between this algorithm and Moore's algorithm which produces optimal schedules when set-up times are sequence independent.

The second algorithm produces optimal schedules when jobs may be processed in any order. A branch and bound technique is developed and illustrated with an example problem. Finally, the amount of computation required by this algorithm is compared to that required for the algorithm of Little, et.al., to solve an n city traveling salesman problem.

II. PROBLEM DESCRIPTION

A single machine is used to process a known set of jobs. Each job has only one operation and its processing time and due-date are known. The machine set-up time for each job is known but is dependent upon the job which precedes it. A job once begun is processed until completion. The measure of performance is to minimize the number of late jobs when

1) Those jobs which are included in the schedule must be processed according to first-come, first-served (FCFS) discipline,

2) jobs may be processed in any order.

The nature and difficulties associated with this problem are discussed in reference (1) where a measure of performance of minimization of maximum flow time is selected and the problem solved as a travelling salesman problem. References (2), (3), and (4) treat the related problem where set-up times are sequence-independent.

III. JOBS PROCESSED IN FCFS ORDER

Throughout this paper the following notation is employed.

p_i = known processing time for job i independent of sequence, $i = 1,\ldots,n$

d_i = known due-date for job i

$s_{i,j}$ = set-up time for job j given that it follows job i, $i = 0,1,\ldots,n$

c_i = completion time for job i

It is convenient to combine the terms p_j and $s_{i,j}$ to form $p_{i,j}$, the total set-up and processing time for job j given that it is immediately preceeded by job i. It is also convenient to add an extra job designated as job zero to represent the preliminary idle condition of the machine. Thus, s_{oj} $j = 1,\ldots,n$ is the set-up time for job j given that job j is processed first.

A. THE FCFS ALGORITHM

The algorithm begins by ordering the jobs in the FCFS sequence. If there are no late jobs, this sequence is optimal. If not, one or more jobs must be excluded from the sequence to produce an optimal processing schedule. Jobs once excluded from the original sequence are not considered again and can be processed in any order after the last job included in the optimal schedule. Suppose job j is the first late job in the original sequence, then from among jobs $1,\ldots,j$ that job is excluded which minimizes the time required to process the remaining jobs. This rule is repeated each time an exclusion is required.

The steps in the algorithm are as follows:

1) Number the jobs according to the order in which they must be processed and form the $(n+1)$ by n matrix $(p_{i,j})$. The element in the 0th row and the jth column represents the processing time plus the set-up time for job j given that job j is processed first.

2) Augment the matrix $(p_{i,j})$ with an additional row which contains the due-dates for the jobs.

3) Compute the completion time c_i, $i=1,2,\ldots k$ where

$$c_i = c_{i-1} + p_{i-1,i}\ , \qquad c_0 = 0\ .$$

For the first k such that $c_k > d_k$, go to step 4. If $c_i \leq d_i$, $i=1,\ldots,n$, then the minimum number of late jobs is zero.

4) For each column 1,2,...,k compute the savings S_i where

$$S_i = c_{i+1} - (c_{i-1}+p_{i-1,i+1})$$

5) From the matrix, delete row and column h such that

$$S_h = \max \{S_i, i=1,...,k\} .$$

Go to step 3 and recompute c_i's starting with job h+1. (Note that after the matrix is reduced, the subscripts refer to relative positions of columns in the matrix rather than job numbers in the computational formulas for c_i and S_i .)

The set of columns in the final matrix specifies the largest set of jobs which can be processed before their due dates. The remaining jobs, which will be late, can be processed in any order following the last early job.

B. EXAMPLE

Let the augmented matrix of step 2 be that shown in figure 1.

	1	2	3	4	5	6	7	8
0	21	13	35	14	29	35	24	32
1	21(20)	12	15	26	11	16	18	20
2	39	33(23)	26	15	16	22	14	20
3		40	59(57)	46	11	11	30	24
4			48	105	30	35	13	24
5				50		5	21	13
6					70		29	15
7						80		11
8							81	
9	39	40	48	50	70	80	81	100

Figure 1: The Augmented Matrix

The elements from row 9 have been moved up under the main diagonal for convenience. Step 3 has been carried out until $c_3 > d_3$. c_i's are shown in the main diagonal spaces. Step 4 has been completed with S_i's shown in parenthesis in the main diagonal spaces. The quantity S_3 is max $\{S_i, i=1,2,3\}$, so delete row and column 3 and return to step 3.

	1	2	4	5	6	7	8
0	21	13	14	29	35	24	32
1	21(20)	12	26	11	16	18	20
2	39	33(1)	15	16	22	14	20
4		40	48(29)	30	35	13	24
5			50	78(0)	5	21	13
6				70	83	29	15
7					80		11
8						81	
9							100

Figure 2: The First Reduced Matrix

The reduced matrix is shown in figure 2 and again steps 3 and 4 have been completed showing max $\{S_i, i=1,2,4,5\}$ to be S_4. Delete row and column 4 and return again to step 3. The resulting reduced matrix is shown in figure 3.

	1	2	5	6	7	8
0	21	13	29	35	24	32
1	21(20)	12	11	16	18	20
2	39	33(17)	16	22	14	20
5		40	49(-1)	5	21	13
6			70	54(13)	29	15
7				80	83(25)	11
8					81	94
9						100

Figure 3: The Second Reduced Matrix

Steps 3 and 4 show now that S_7 is maximum, so proceeding as before, delete row and column 7.

	1	2	5	6	8
0	21	13	29	35	32
1	21	12	11	16	20
2	39	33	16	22	20
5		40	49	5	13
6			70	54	15
8				80	69
9					100

Figure 4: The Solution Matrix

It can be seen in figure 4, after deleting row and column 7 and carrying out step 3 that now all remaining $c_i \leq d_i$ and the

optimal schedule is (1,2,5,6,8). This solution also has the property that for all solutions in which only 3 jobs are processed late, this permutation completes processing of the early jobs in minimum time.

C. PROOF OF OPTIMALITY

As is done in references (3) and (4), jobs will be separated into two disjoint sets, E and L, corresponding to those jobs which are processed early and those which are not according to the current schedule. Define

J_i = i^{th} job in FCFS sequence of jobs.

E_k = those jobs out of the first k which have been retained in the processing schedule.

L_k = those jobs out of the first k which have been excluded from the processing schedule.

$|E_k|$ = the number of jobs in set E_k. (Similarly for $|L_k|$).

Given that out of the first k jobs in the sequence $|L_k|$ must be excluded, then it is optimal to place those jobs in L_k which allow the $|E_k|$ jobs in E_k to be completed in minimum time since this will allow the remaining n-k jobs the greatest opportunity to be completed before their due-dates.

The optimality of this criterion can be seen by supposing that for some current schedule with $|L_k| = j$, $2 \leq k < n$, the next job in the sequence has $c_{k+1} < d_{k+1}$. In this case $|L_{k+1}| = j$ and c_{k+1} is minimal. On the other hand, if $c_{k+1} > d_{k+1}$, then

$|L_{k+1}| = j+1$. The job to be placed in L_{k+1} is job J_r where J_r is in $E_k \cup \{J_{k+1}\}$ and $S_r = \max_i \{S_i : J_i \text{ in } E_k \cup \{J_{k+1}\}\}$. Then $c_{k+1} - S_r \leq c_{k+1} - S_i$, J_i in $E_k \cup \{J_{k+1}\}$. Hence, c'_k, the adjusted time to complete job k, given that one more job has been placed in L_k, is minimized by selecting J_r for placement in L_k. Optimality is thus proved.

D. COMPARISON TO MOORE'S ALGORITHM

If $s_{1,j} = s_{2,j} = \ldots = s_{n,j} = s_j$, i.e., if the set-up times for all jobs are independent of sequence, then

$p_{i,j} = p_j + s_{i,j} = p_j + s_j$, is the combined constant set-up and processing time for job j. The elements in the j^{th} column of the problem matrix will now all be equal. Hence, $S_i = p_i + s_i$ for each i independent of sequence. The algorithm presented in this paper will choose that job out of the first k with the largest processing time and place it in L, as does Moore's algorithm.

IV. NO RESTRICTION ON PROCESSING SEQUENCE

In this formulation all n jobs are assumed to be simultaneously available for processing and no restrictions are imposed on the processing sequence. This algorithm uses a problem matrix similar to that prescribed by step 2 of the FCFS algorithm just described except now both the upper and lower triangles will be required since job numbered (i+1) may preceed job i.

A branch and bound technique is developed in which branching to a node labeled i means "process job i next." The bound computed at each node is a lower bound on the minimum number of jobs which will be processed late if the solution set represented by that node is used.

A. THE BRANCH AND BOUND ALGORITHM

Steps 1 and 2 are the same as in the FCFS algorithm but are repeated here for reference.

1) Number the jobs in non-decreasing due-date order and form the $(n+1)$ by n matrix (p_{ij}). The element in the 0^{th} row and the j^{th} column represent the processing time plus the set-up time for job j given that job j is processed first.

2) Augment the matrix $(p_{i,j})$ with an additional row which contains the due-dates for the jobs.

3) At the beginning node designated "ALL", compute a first bound B on all solutions by comparing $p_{0,j}$ to d_j, $j=1,\ldots,n$. For each j such that $p_{o,j} > d_j$, increment B by 1 and delete the corresponding row and column. (This step removes from consideration any jobs which cannot be processed on time no matter what their position in the sequence.)

4) Branch from the current node labelled k to each remaining job as follows: For job j, subtract $p_{k,j}$ from

all elements in row (n+1) and compare the resulting elements in row (n+1), call them ($d_i(j)$, to the corresponding elements in row j. For each $p_{j,i} > d_i(j)$, increment the bound on node j by 1.

5) Branch now as described in step 4 from the node with the least bound. If there is a tie, branch from the node with minimum d_i. If there still is a tie, branch from the node among the tied nodes with minimum $p_{i,j}$. Repeat step 5 until branching is completed. An optimal sequence is described by tracing back through the solution tree.

B. EXAMPLE

The problem matrix is as shown in figure 5.

	1	2	3	4	5
0	21	13	14	29	32
1	0	11	25	17	27
2	24	0	15	16	20
3	45	22	0	30	24
4	50	36	44	0	13
5	64	41	33	27	0
6	20	40	50	70	80

Figure 5: Problem Matrix

Carrying out step 3, obtain a first bound on all solutions by comparing row 0 to row 6. Note $p_{0,1} > d_1$, so set B=1 for node "ALL" and delete row and column 1.

C. DISCUSSION

It may appear that a relatively large number of bounds must be computed as compared to well known algorithms (5) presently used to solve the very similar traveling salesman problem. This may turn out to be the case. However, the algorithm takes great advantage of one of the trade-offs available in branch and bound methods in that bounds are very easily calculated. The possible requirement to compute more bounds then does not necessarily imply reduced efficiency. In fact, for the example problem given only $2\sum_{i=1}^{n-2}(n-i)^2 + n = 63$ additions and comparisons were required to solve the problem. An n-city traveling salesman problem solved by the algorithm of Little, et. al., reference (5), would require more than $4\sum_{i=1}^{n-2}(n-i)(n-i-1) + 3n(n-1) = 140$ additions and comparisons even if branching were only required directly down a single branch. The numbers of computations indicated above represent lower bounds for each algorithm. For the algorithm presented here, if branching had been required into the second level from each of the nodes at level 1 and the remaining computations completed as in the example, the required number of additions and comparisons would have been approximately $2(n-1)\sum_{i=1}^{n-2}(n-i)^2 + n = 237$. No firm conclusion can be drawn from these comparisons, but they are presented as an indication that the computational efficiency of this algorithm deserves further investigation.

REFERENCES

[1] Conway, R. W., Maxwell, W. L. and Miller, L. W., *Theory of Scheduling*, Addison, Wesley, Reading, Mass., 1967.

[2] Maxwell, William L., "On Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs," *Management Science*, Vol. 16, No. 5, January 1970.

[3] Moore, J. M., "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," *Management Science*, Vol. 15, No. 1, September 1968.

[4] Emmons, H., "A Simplified Algorithm for Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs," Technical Report No. 51, Dept. of Operations Research, Cornell University, Ithaca, New York, August 1958.

[5] Little, J. D. C., Murty, G., Sweeney, D. W. and Karel, C., "An Algorithm for the Traveling Salesman Problem," *Operations Research*, 11, 972-989 (1963).

INITIAL DISTRIBUTION LIST

	No. of Copies
Advanced Research Projects Agency Department of Defense Washington, D.C. 20301 (Technical Library)	1
National Security Agency Ft. Meade Maryland 20755	1
National Military Command System Support Center Defense Communications Agency Washington, D.C. 20301	1
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Chief of Naval Operations Department of the Navy Washington, D.C.	
(Op-96)	1
(Op-94)	1
(Op-91)	1
Office of Naval Research Arlington, Virginia 22217	
(Code 462)	2
(Code 435)	1
(Code 432)	1
U.S. Marine Corps Headquarters (Code AX) Washington, D.C. 20380	1
Commander Submarine Development Group TWO Naval Submarine Base New London Box 70 Groton, Connecticut 06340	2
Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101	1

Naval Academy 1
Annapolis, Maryland 21402
(Technical Library)

Naval War College 1
Newport, Rhode Island 02840

Naval Research Laboratory
Washington, D.C. 20390
(Technical Information Division) 6
(Code 2629) 2

Naval Ordnance Laboratory
Silver Spring, Maryland 20910
(Code 120) 1
(Library) 1

Naval Weapons Laboratory
Dahlgren, Virginia 22448
(Code KCU) 1
(Code KWP) 1

Naval Air Development Center (SAED) 1
Warminster, Pennsylvania 18974

Naval Ship Research and Development Center 1
Washington, D.C. 20034

Navy Underwater Systems Center 1
Newport Laboratory
Newport, Rhode Island 02840

Naval Undersea Research and Development Center 1
San Diego, California 92152

Naval Undersea Research and Development Center 1
Pasadena Laboratory
3202 East Foothill Boulevard
Pasadena, California 91107

Naval Weapons Center
China Lake, California 93555
(Library) 1
(Code 12) 1

Center for Naval Analyses 1
1401 Wilson Boulevard
Arlington, Virginia 22209
(Library)
(Surfact Section (Seacad Div. of OEG)) 1

HRB-Singer Company
P.O. Box 60
State College, Pennsylvania 16801
(Mr. D. Smith) — 1

Institute of Defense Analyses
400 Army Navy Drive
Arlington, Virginia 22202 — 1

The Ohio State University
1775 South College Road
308 Hagerty Hall
Columbus, Ohio 43210
(Dr. D. Howland) — 1

Operations Research Incorporated
1400 Spring Street
Silver Spring, Maryland 20910 — 1

Stanford University
Department of Engineering-Economic Systems
Stanford, California 94305 — 1

Daniel H. Wagner Associates, Inc.
Station Square One
Paoli, Pennsylvania 19301 — 1

University of Wisconsin
Army Mathematics Research Center
Madison, Wisconsin 53706 — 1

Professor G. T. Howard
Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940 — 10

Library (Code 0212)
Naval Postgraduate School
Monterey, California 93940 — 2

Dean of Research Administration
Code 023
Naval Postgraduate School
Monterey, California 93940 — 2

Stephen J. Balut, LCDR, USN
Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940 — 10

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3 REPORT TITLE

N Job, One Machine Scheduling to Minimize the Number of Late Jobs When Set-Up Times are Sequence Dependent

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Gilbert T. Howard
Stephen J. Balut

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1972	22	5

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

Two algorithms have been formulated for scheduling n jobs through a single facility to minimize the number of late jobs when set-up times are sequence dependent. The first is a simple matrix algorithm which solves the problem when jobs must be processes in first-come, first-served (FCFS) order. The second is a branch and bound technique which arrives at an optimal solution with no restrictions on the sequence used. Both algorithms are demonstrated by examples.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Scheduling						
Set up time						
Job shop scheduling						